Amendments to the Specification:

Please amend the paragraph at page 6 lines 8 through 12 as follows:

In Fig. 5, in accordance with the invention, an address trace cache 220 is composed of a start address for storing an address where each routine is started, an end address for an address where each routine is finished, a current loop iteration ~~loop~~ counter for ~~counting the iterations a~~ generating a current iteration count of a corresponding routine ~~to generate a current iteration count~~, and a total loop iteration ~~loop~~ count~~er~~ for indicating total number of iterations of the routine. That is, the total loop iteration count is a stored value indicating the total number of iterations of the routine to be executed.

Please amend the paragraph at page 6, lines 13 through 20 as follows:

For example, if the information of instructions executed in routine 1 is indicated the address trace cache 220, the start address and the end address of routine 1 are stored in the trace cache 220. Then, current iteration count of routine 1 is stored in the current iteration loop counter while total number of iterations (e.g., 30 times) of routine 1 is stored in the total loop iteration ~~loop~~ count~~er~~. As execution of the routine is repeated, a value of the current loop iteration ~~loop~~ counter is increased. If the value of the current loop iteration ~~loop~~ counter is identical to that of the total loop iteration ~~loop~~ count~~er~~, routine 1 is finished and a start address of routine 2 is stored as a next fetch point (NFP).

Please amend the paragraph at page 6 line 27 through page 7 line 4 as follows:

Referring now to Fig. 6, if, for example, a routine 1 shown in Fig. 3 is stored in the address trace cache 220, an address of an initially executed instruction A is stored into a start address of routine 1 while an address of a finally executed instruction B is stored into an end address thereof. Since the total number of iterations or repetitions of the routine 1 is 30, the value of the total loop iteration ~~loop~~ count~~er~~ is stored as 30. Whenever routine 1 is repeatedly carried out, a value of the current loop iteration ~~loop~~ counter increases by 1. In the same manner, information of routines 2 and 3 is stored in the address cache 220.

Please amend the paragraph at page 7 lines 5 through 12 as follows:

As shown in Fig. 6, since the address cache 220 is composed of an address where each routine is started, an address where each routine is finished, a current loop iteration ~~loop~~ counter, and a total loop iteration counter, only four data storing areas are required to store information of a repeated routine. Therefore, a total of 12 data storing areas are required to store routines 1 through 3 in the address trace cache 220. In this case, if 32 bits are utilized to store each piece of information, a total of 384 bits (i.e., 48 bytes) are required to store routines 1 through 3. This is smaller by about 16.7 times than a data storing area utilized in a conventional trace cache.

Please amend the Abstract of the Disclosure at page 9 lines 1 through 9 as follows:

A branch prediction method using an address trace is described. An address trace corresponding to an executed instruction is stored itself with a decoded form. After appointing a start address and an end address of a repeated routine, current routine iteration count ~~accessing times~~ and total ~~accessing times~~ number of iterations are compared with each other, confirming the end of the routine and storing address information of the next routine. Therefore, access information of the repeated routine can be stored using a small amount of ~~a~~ trace cache.